04

Getter

Transcripción

[00:00] ¿Qué método podría ser? Un método como este de aquí, por ejemplo tengo un método para depositar, un método para retirar, entonces yo puedo crear aquí abajo un método, digamos, obtener saldo. Cuando yo escribo un método, repasando una clase anterior, escribo el nombre en camel case, por convención de Java, uso mis paréntesis, abro mis llaves y cierro el scope de mi método.

[00:34] En este caso mi saldo es una variable del tipo double, entonces aquí como yo voy a obtener mi saldo, yo voy a devolver un double. Y este método va a ser un método público. ¿Para qué? Para que sea accesible desde todas partes del código. Y aquí yo hago algo tan simple como retornar. ¿Qué voy a retornar? Mi saldo.

[01:10] El saldo de esta cuenta, de este objeto sería this.saldo, punto y coma, y de esta forma vemos que el código que no compila cuenta.saldo ahora podría ser cuenta.obtenerSaldo. Pero en este caso tampoco está compilando. ¿Por qué? Porque obtener saldo es un método que nosotros usamos para acceder al saldo y obtener el valor de esa variable, no para igualarla a un valor.

[01:50] Para este caso, si nosotros deseamos modificar el saldo de esta cuenta, existe otro tipo de método, podríamos crear otro método para definirle un valor. Pero eso es justamente lo que queremos evitar, queremos evitar que intervengas directamente con la variable entonces esto ya no va a tener más sentido por el momento.

[02:13] Igual aquí, esto tampoco va a tener sentido por el momento, porque no queremos intervenir directamente con la variable. Entonces borramos aquí, y aquí nosotros sí queremos saber el saldo. Y aquí sí le decimos obtenerSaldo. Y de esta forma nosotros vamos a obtener el saldo actual de esa cuenta.

[02:39] Vamos aquí, ejecutamos, guardamos. Aquí dice que hay errores en el proyecto. ¿Por qué? Vamos a decirle igual que proceda pero vamos a ver después esos errores. Y vemos aquí que el saldo es cero. ¿Por qué? Porque en este caso yo o le he definido un saldo aún. El saldo acuérdense que el valor con el que está inicializando en new cuenta es cero.

[03:10] Por lo tanto, si yo deseo retirar 300, él va a hacer la validación aquí. Va a decir: "Ah, no, tú no puedes retirar, va a retornar un falso y no va a hacer nada con mi saldo". De esta forma mi saldo ya está protegido, ya está tranquilo ya que nadie va a intervenir directamente con él.

[03:32] Si yo quisiera probar si obtener saldo de verdad es atrayendo ese valor, entonces vamos a decirle aquí, antes de retirar. ¿Qué vamos a decirle? Depositar. ¿Cuánto vamos a depositar? Vamos a depositar por ejemplo 400, entonces cuenta, depositamos 400, retiramos 300 y obtenemos el saldo que sería ¿cuánto? Sería igual a 100.

[04:03] ¿A qué se refiere este error que me está saliendo aquí cada vez que yo estoy ejecutando el código? Este error es aquí en los archivos que no está compilando. ¿Por qué? Porque saldo ya no es accesible desde ninguna parte del código. Saldo está encapsulado, saldo ya está escondido, por lo cual en este caso si yo pongo ya un obtenerSaldo, ya va a compilar tranquilamente.

[04:34] En el caso de Eclipse, como este método es totalmente aislado del resto del proyecto, yo puedo seguir aquí ejecutando misión métodos, pero de todas formas yo tengo aquí que actualizar mi código para obtener el saldo ya a través del método que he creado.